VII. Conclusion

A technique has been presented for implementation of residue multipliers for non-prime moduli, where the moduli can be decomposed into two or more relatively prime factors. An all ROM table look-up implementation of this technique was considered and analysis of the hardware requirements showed that the size of multipliers for non-prime moduli was less for a large proportion of moduli than that of prime moduli of comparable magnitude. An advantage of this multiplication technique in comparison with the quarter square multiplier is that the final stage is a look-up table, allowing the inclusion of another "free" operation in the final stage. Finally it was shown that the introduction of further stages of modulus decomposition can be implemented resulting in large hardware savings for some moduli.

Acknowledgment

The author is grateful for the use of the facilities of the VaST and JMRC laboratories at the University of New South Wales and for helpful discussions with Professor Graham Rigby.

References

[1] M. A. Soderstrand and C. Vernia, "A high-speed low-cost modulo P_i multiplier with RNS arithmetic applications," *Proc. IEEE*, vol. 68, pp. 529–532, Apr. 1980.
[2] W. K. Jenkins, "Complex residue number arithmetic for high-speed signal processing," *Electron. Lett.*, vol. 16, no. 17, pp. 660–661, Aug. 1980.
[3] G. A. Jullien, "Implementation of multiplication modulo a prime number with applications to number theoretic transforms," *IEEE Trans. Comput.*, vol. C-29, pp. 899–905, Oct. 1980.
[4] F. J. Taylor, "Large moduli multipliers for signal processing," *IEEE Trans. Circ. Syst.*, vol. CAS-28, pp. 731–736, July 1981.
[5] D. Radhakrishnan and Y. Yuan, "A fast RNS Galois field multiplier," in *Proc. IEEE ISCAS 1990*, pp. 2909–2912, May 1990.
[6] H. C. Shyu, T. K. Truong, and I. S. Reed, "A complex integer multiplier using the quadratic-polynomial residue number system with numbers of form $2^{2n}+1$," *IEEE Trans. Comput.*, vol. C-36, no. 10, Oct. 1987.
[7] I. S. Reed and T. K. Truong, "VLSI residue multiplier modulo a Fermat number," in *Proc. 8th Symp. on Comput. Arith.*, Urbana, IL, June 1985.
[8] F. J. Taylor, "A VLSI residue arithmetic multiplier," *IEEE Trans. Comput.*, vol. C-31, no. 6, June 1982.
[9] G. Alia and E. Martinelli, "A VLSI modulo m multiplier," *IEEE Trans. Comput.*, vol. 40, no. 7, July 1991.
[10] K. M. Elleithy and M. A. Bayoumi, "A $\Theta(\log n)$ algorithm for modulo multiplication," in *Proc. 1990 Asilomar Conf. Circuits, Syst., Comput.*, Pacific Grove, CA, 1990.

Generalized Polyphase Representation And Application To Coding Gain Enhancement

Anand K. Soman, and P. P. Vaidyanathan

***Abstract*— Generalized polyphase representations (GPP) have been mentioned in literature in the context of several applications. In this paper, we provide a characterization for what constitutes a valid GPP. Then, we study an application of GPP, namely in improving the coding gains of transform coding systems. We also prove several properties of the GPP.**

I. Introduction

The polyphase representation is a useful tool in multirate applications [1]-[3], [11]. It has been extensively used in the design of digital filter banks. The M-fold polyphase representation of a transfer function $H(z)$ is given by

$$H(z) = \sum_{i=0}^{M-1} h_i(z^M)z^{-i}, \tag{1}$$

where the $h_i(z)$ are referred to as the M polyphase components of $H(z)$. The right hand side of (1) is a linear combination of functions $z^{-i}, \quad i = 0, \ldots, M-1$, with the weighting factors being functions of z^M. Such a representation holds for both finite and infinite impulse response (FIR and IIR) transfer functions. Moreover, $H(z)$ is FIR if and only if all its polyphase components are FIR. A natural question which arises is whether an arbitrary transfer function $H(z)$ may be written as a linear combination of functions other than z^{-i}, while retaining the desirable properties [1]-[3] of the traditional polyphase representation. Furthermore, are there any advantages to be gained by using a different set of functions?

In [3], the author has mentioned the so called 'generalized polyphase representation' (GPP). It has been shown that using a GPP, it is possible to efficiently quantize the coefficients of a digital filter. It has also been shown therein that the GPP gives a second derivation of the so called Interpolated FIR (IFIR) filter technique [8]. In [4], further applications of GPP have been studied. However, neither of these references addresses the issue of what constitutes a valid generalized polyphase representation. In this paper we first provide a complete characterization of valid polyphase representations (Section II). In Section III, we study another application of the GPP, namely in enhancing the coding gain of transform coding systems. We prove several interesting properties in this regard.

The notation used in this paper closely follows that used in [3]. Bold faced quantites denote vectors and matrices. Let $x(n)$ be a real, wide sense stationary (WSS) random process. The correlation function $\rho(k)$ of this process is defined as $\rho(k) = E[x(n)x(n-k)]$. If $\boldsymbol{x}(n)$ is a WSS vector random process, its M by M autocorrelation matrix is defined as $\boldsymbol{R_{xx}}(k) = E[\boldsymbol{x}(n)\boldsymbol{x}^T(n-k)]$. AR(N) refers to an autoregressive process of order N [6]. The abbreviation gcd stands for 'greatest common divisor'. In the figures, the boxes with $\uparrow M$ and $\downarrow M$ stand for interpolators and decimators respectively, as defined in [2], [3].

Manuscript received April 23, 1992; revised March 10, 1993. This paper was recommended by Associate Editor M. A. Sonderstrand. This work was supported in parts by NSF grant MIP 8919196, Tektronix Inc., and Rockwell International.

The authors are with the Department of Electrical Engineering, 116-81, California Institute of Technology, Pasadena, CA 91125 USA.

IEEE Log Number 9403068.

II. Generalized Polyphase Representations

In this section we first define a 'valid polyphase representation' (VPP) and then provide a characterization of all such representations.

Definition: Let $\boldsymbol{u}(z) = [u_0(z)\ u_1(z)\ \dots\ u_{M-1}(z)]^T$. This is said to be a valid polyphase representation (VPP) if (a) every rational function $B(z)$ can be represented as $B(z) = \sum_{i=0}^{M-1} b_i(z^M)u_i(z)$, where the $b_i(z)$ are rational (b) $b_i(z)$ are FIR if and only if $B(z)$ is FIR.

It can be shown that with this definition, $\boldsymbol{u}(z)$ is guaranteed to be FIR.

We now characterize all such VPPs. Let $\boldsymbol{e}(z) = [1\ z^{-1}\ z^{-2}\ \dots\ z^{-M+1}]^T$. This is, therefore, the basis for the usual polyphase representation. Let the vector $\boldsymbol{u}(z)$ defined above be given the usual polyphase representation $\boldsymbol{u}(z) = \boldsymbol{V}(z^M)\boldsymbol{e}(z)$. This means that $\boldsymbol{V}(z)$ is the conventional polyphase matrix [3] of the elements of the vector $\boldsymbol{u}(z)$. Note that $\boldsymbol{V}(z)$ is FIR. We have the following result:

Lemma 2.1: $\boldsymbol{u}(z)$ is a valid polyphase basis if and only if $\det[\boldsymbol{V}(z)] = cz^k$ for $c \neq 0$ and integer k.

Proof: First assume that $\boldsymbol{u}(z)$ is a VPP. Then every transfer function can be represented in terms of the elements of $\boldsymbol{u}(z)$. In particular, $\boldsymbol{e}(z)$ can be written in terms of $\boldsymbol{u}(z)$ as

$$\boldsymbol{e}(z) = \boldsymbol{E}(z^M)\boldsymbol{u}(z). \tag{2}$$

But, $\boldsymbol{u}(z) = \boldsymbol{V}(z^M)\boldsymbol{e}(z)$. Hence

$$\boldsymbol{e}(z) = \boldsymbol{E}(z^M)\boldsymbol{V}(z^M)\boldsymbol{e}(z). \tag{3}$$

Now, $\boldsymbol{E}(z)\boldsymbol{V}(z)$ is the traditional polyphase matrix of $\boldsymbol{e}(z)$ with respect to $\boldsymbol{e}(z)$. Therefore, $\boldsymbol{E}(z)\boldsymbol{V}(z) = \boldsymbol{I}$. Since, $\boldsymbol{E}(z)$ and $\boldsymbol{V}(z)$ are both FIR, we have the result that $\det[\boldsymbol{V}(z)]$ is a power of z.

Conversely, let $\det[\boldsymbol{V}(z)]$ be a power of z. We know that any transfer function $H(z)$ can be represented as $H(z) = h^T(z^M)\boldsymbol{e}(z)$. Using (2) this becomes $H(z) = h^T(z^M)\boldsymbol{E}(z^M)\boldsymbol{u}(z)$. Hence $H(z)$ can be represented in terms of $\boldsymbol{u}(z)$. Since $\boldsymbol{V}(z)$ is FIR and $\det[\boldsymbol{V}(z)]$ is a power of z, $\boldsymbol{E}(z^M)$ should be FIR (using $\boldsymbol{E}(z^M)\boldsymbol{V}(z^M) = \boldsymbol{I}$). Hence $h^T(z^M)\boldsymbol{E}(z^M)$ is also FIR for FIR $H(z)$. This proves the converse. □

III. Coding Gain Enhancement Using GPP

In this section, we shall study a specific application of the generalized polyphase representation.

Consider Fig. 1 with $J_i = 1, i = 1, \dots, M-1$. This is therefore the familiar case of Transform coding. Such schemes are used in data compression of speech, images and other signals. In such a scheme, the input string is divided into non-overlapping blocks $\boldsymbol{x}(n)$ of length M by grouping together M successive samples. Each block is encoded by multiplying it with a transform matrix $\boldsymbol{A}$. The transform coefficients $\boldsymbol{s}(n)$ are independently quantized. At the receiver, the inverse transformation $\boldsymbol{A}^{-1}$ is applied to the received vector $\boldsymbol{t}(n)$ to produce the output vector, which is 'unblocked' to obtain the output sequence. The case where the transform matrix is orthogonal $(\boldsymbol{A}^T = \boldsymbol{A}^{-1})$ is called Orthogonal Transform Coding [6], and is the one most commonly used in practice.

There are two issues involved in the design of transform coding systems; namely, allocating the bits to the individual quantizers, and choosing the 'optimal' transform matrix $\boldsymbol{A}$ so as to maximize the coding gain. The optimal bit allocation result [6] says that the distribution of bits which minimizes the reconstruction error variance is the one that makes the individual quantizer error variances equal. Also, it is well known that the transform matrix $\boldsymbol{A}$ which maximizes the coding gain of the system is the Karhunen-Loeve Transform (KLT), whose rows are the eigenvectors of the input autocorrelation matrix [6]. The coding gain then becomes

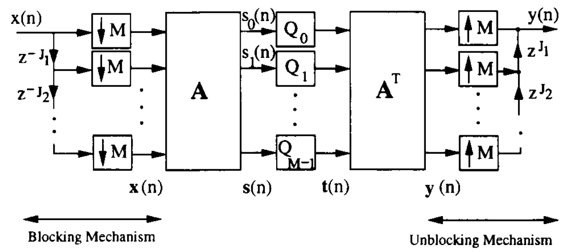

Fig. 1. A generalized transform coding system.

$$G_{TC} = \frac{\sigma_x^2}{(\det[\boldsymbol{R_{xx}}(0)])^{1/M}}. \tag{4}$$

An aspect of the transform coding scheme which has not received attention so far is the variations of the blocking/unblocking mechanisms (Fig. 1). Notice that in a traditional transform coding system, this mechanism is responsible for blocking M *successive* samples of the input data. However, it is possible in case of certain inputs to exploit the correlations between non-adjacent samples of the input data so as to enhance the coding gain. This would be particularly important when data from several sources is multiplexed into one bit-stream. Specifically, it is possible in several cases to design the blocking mechanism such that the value of $(\det[\boldsymbol{R_{xx}}(0)])$ in (4) is reduced. The question now is what are the constraints which the new blocking/unblocking mechanism has to satisfy?

Consider the system shown in Fig. 1 with $J_i = J, i = 1, \dots, M-1$. Hence, we have used a generalized polyphase basis comprising of the functions z^{-iJ}, $i = 0, \dots, M-1$. The matrix $\boldsymbol{A}$ is the polyphase matrix of the filters in a generalized sense. Since the basis can be implemented using only delay elements, this scheme is equivalent to a transform coding scheme in terms of complexity.

Fact 3.1: Consider the system shown in Fig. 1 with $J_i = J$, $i = 1, \dots, M-1$. This is a perfect reconstruction system if and only if $gcd(J, M) = 1$.

Proof: One proof of this fact appears in [9]. We present here a proof based on GPP for the sake of completeness. Let $\boldsymbol{u}(z)$ be the vector with elements $u_i(z) = z^{-iJ}$, $i = 0, \dots, M-1$. Let $\boldsymbol{u}(z) = \boldsymbol{V}(z^M)\boldsymbol{e}(z)$, where $\boldsymbol{e}(z)$ is as defined above. If $gcd(J, M) = 1$, then it can be verified that the matrix $\boldsymbol{V}(z)$ has only one entry per column, and this entry is a delay. Hence $\det[\boldsymbol{V}(z)]$ is a delay, implying that this is a valid GPP. Furthermore, $\boldsymbol{V}(z)$ is paraunitary [3], i.e., it satisfies $\boldsymbol{V}(z)\widetilde{\boldsymbol{V}}(z) = \boldsymbol{I}$, where $\widetilde{\boldsymbol{V}}(z)$ is obtained from $\boldsymbol{V}(z)$ by transposition, followed by conjugation of coefficients followed by replacing z by z^{-1}. The polyphase matrix of the unblocking mechanism is $\widetilde{\boldsymbol{V}}(z)$, and hence the system is a perfect-reconstruction system. Conversely, if $gcd(J, M) \neq 1$, it can be verified that at least one of the columns of $\boldsymbol{V}(z)$ will have all zeros, and hence the system cannot have perfect reconstruction property. □

Comment: Suppose M is fixed. There are several choices of J which satisfy Fact 3.1. In practice, we choose J such that the correlation between samples distance J apart is high. If the selection of both J and M is upto the designer, J is first chosen as above, and then M is chosen so as to satisfy Fact 3.1. However, we have not proved theoretically the optimality of such an approach.

Coding gain example: As an example of a process where the coding gain of the new system is better than the transform coding system, consider a process with the autocorrelation function

$$\begin{aligned}\rho(k) &= \rho_1(k) + \rho_2(k) \quad \text{where}\\ \rho_1(k) &= (0.1)^{|k|} \quad \text{and}\\ \rho_2(k) &= (0.9)^{|k|/4} \quad \text{if } |k| \text{ is a multiple of 4,}\\ &\qquad \text{and 0 otherwise.}\end{aligned} \tag{5}$$

Such an autocorrelation could arise where for example, the correlation between non-adjacent samples is high. If we used a traditional transform coding scheme on such an input, the coding gain would only be 0.029 db, whereas using $J = 4$ gives a gain of 1.63 db, (in both cases, $M = 3$).

Transform coding is often used to encode images. Data from images normally shows high correlation between adjacent samples, and is often modelled as an AR(1) process. For such data, the choice of J and M is simplified by the following lemma.

Lemma 3.2: For an AR(1) process, the value of J which maximizes the coding gain of the system is $J = 1$ (i.e. traditional transform coding), for all M.

Proof: Consider running the Linear Predictive Algorithm (LPC) [6] on the AR(1) input. Let ϵ_i denote the prediction error variances for the ith order optimal predictor. Then it can be shown that $\epsilon_0 = 1$, and $\epsilon_i = (1 - \rho^2)$, $\quad for\ all\ i \geq 1$. Here, ρ is the correlation coefficient of the AR(1) process. Now consider the autocorrelation matrix $\boldsymbol{R_{xx}}$ of size $M \times M$ corresponding to the vector input sequence $\boldsymbol{x}(n)$. It can be shown [6] that the determinant of this matrix is given by $\det[\boldsymbol{R_{xx}}] = \prod_{i=0}^{M-1} \epsilon_i = (1 - \rho^2)^{M-1}$. If we use $J \neq 1$, it can be verified that the autocorrelation matrix of the new vector process is similar to $\boldsymbol{R_{xx}}$, but with correlation coefficient equal to ρ^J. The determinant of the new autocorrelation matrix is $\det[\boldsymbol{R_{xx}}'] = (1 - \rho^{2J})^{M-1}$. If $|\rho| \leq 1$, $\det[\boldsymbol{R_{xx}}'] \geq \det[\boldsymbol{R_{xx}}]$. Hence, from (4), the coding gain of the new system can be no better than the traditional transform coding system. □

Monotonicity: In the case of traditional transform coders, the optimal coding gain can be shown (Appendix C of [10]) to be a monotonic function of the number of channels M for arbitrary inputs. In systems such as in Fig. 1 however, the optimal coding gain is a function of M as well as J. For a given input, and a certain number of channels, there exists a optimal J satisfying Fact 3.1, which maximizes the coding gain of the system. Let $G_{opt}(J_{opt}, M)$ denote the maximum gain after having chosen the optimal J for a particular input. It can be shown that $G_{opt}(J_{opt}, M)$ is *not* a monotonic function of M. To see this, consider the following autocorrelation function

$$R(k) = (\rho)^{|k|} \quad \text{if } k \text{ } is\ a \text{ multiple of 6. and 0 otherwise.} \tag{6}$$

If $M = 5$, and if we use $J = 6$, we get a coding gain of $-\log(1 - \rho^2)^{4/5} db$. If $\rho = 0.95$ for example, this value is 8.08db. However, if $M = 6$, it is not possible to get a coding gain greater than 0 db.

Fact 3.3: Consider the system in Fig. 1. Let $P_k = \sum_{i=1}^{k} J_i$, with $P_0 = 0$. Then the system is a perfect-reconstruction system if and only if the numbers $(P_k) mod M$ are distinct.

Proof: As in the proof of fact 3.1, let $\boldsymbol{u}(z)$ be the vector with elements $u_i(z) = z^{-P_i} \quad i = 0, \ldots, M-1$, and let $\boldsymbol{u}(z) = \boldsymbol{V}(z^M)\boldsymbol{e}(z)$. It can again be verified that under the condition that the numbers $(P_k)\ mod\ M$ are distinct, $\det[\boldsymbol{V}(z)]$ is a delay, implying that this is a valid GPP. Furthermore $\boldsymbol{V}(z)$ is also paraunitary, thereby implying perfect reconstruction. Conversely, if the P_k do not satisfy the stated property, $\boldsymbol{V}(z)$ will be singular, implying that perfect-reconstruction is not possible. □

The important point to note in this new scheme is that the autocorrelation matrix of the vector $\boldsymbol{x}(n)$, i.e., $\boldsymbol{R_{xx}}$ is no longer Toeplitz. Hence it is in general difficult to find the J_i which maximize the coding gain for a given process. This would involve minimization of the determinant of a general positive definite matrix under the constraints imposed by Fact 3.3.

One can, however, construct examples to demonstrate an improvement in the coding gain by using systems such as those in Fig. 1.

Example 1: Let $M = 3$, and consider an AR(5) process at the input whose first six autocorrelation coefficients are $\rho_0 = 1.0$, $\rho_1 = 0.2$, $\rho_2 = -0.45$, $\rho_3 = 0.38$, $\rho_4 = 0.7, \rho_5 = -0.4$. Traditional transform coding would give a gain of 0.5 db, whereas using $J_1 = 4$ and $J_2 = 1$ in Fig. 1 would give a coding gain of 2.3 db.

Example 2: Let $M = 4$, and consider an AR(6) process at the input whose first seven autocorrelation coefficients are $\rho_0 = 1.0$, $\rho_1 = -0.2$, $\rho_2 = -0.2$, $\rho_3 = 0.5$, $\rho_4 = -0.46$, $\rho_5 = 0.39$ $\rho_6 = 0.76$. Traditional transform coding would give a gain of 0.512 db, whereas using $J_1 = 1$, $J_2 = 2$ and $J_3 = 3$ in Fig. 1 would give a coding gain of 3.19 db.

Note: One can verify that the above two examples present valid autocorrelation sequences. This can be done by verifying that the relevant Toeplitz autocorrelation matrices (of size 6×6 in Example 1, and of size 7×7 in Example 2) are positive definite.

In the denominator of coding gain expressions, $\det(\boldsymbol{R_{xx}})$ plays a crucial role. So it is important to explore the meaning of $\det(\boldsymbol{R_{xx}}) = 0$. In the traditional case, we know that the the $M \times M$ matrix $\boldsymbol{R_{xx}}(0)$ is singular if the input process $x(n)$ is harmonic with atmost M frequencies. In the case of the system shown in Fig. 1, the following result holds:

Lemma 3.4: Consider the system in Fig. 1, and let $P_k = \sum_{i=1}^{k} J_i$ with $P_0 = 0$. Let the $M \times M$ autocorrelation matrix $\boldsymbol{R_{xx}}$ be singular. Then, the input process $x(n)$ is harmonic with atmost P_{M-1} frequencies.

Proof: Let $\boldsymbol{R}_B$ be the autocorrelation matrix of size $(P_{M-1} + 1) \times (P_{M-1} + 1)$ corresponding to the input sequence $x(n)$, i.e. $\boldsymbol{R}_{B_{i,j}} = [\rho(i-j)]$. We know [7] that if $\boldsymbol{R}_B$ is singular, the input process is harmonic with atmost P_{M-1} frequencies. We now show that $\det[\boldsymbol{R}_B] \leq \det[\boldsymbol{R_{xx}}]$. Since both autocorrelation matrices are positive semi-definite, singularity of $\boldsymbol{R_{xx}}$ would guarantee the singularity of $\boldsymbol{R}_B$.

For a suitable choice of permutation matrix $\boldsymbol{P}$, we have

$$\boldsymbol{P}\boldsymbol{R}_B\boldsymbol{P} = \boldsymbol{Q} = \begin{pmatrix} \boldsymbol{R_{xx}} & \boldsymbol{X} \\ \boldsymbol{X}^T & \boldsymbol{Y} \end{pmatrix}. \tag{7}$$

Hence (pg. 404 of [12]),

$$\det[\boldsymbol{R}_B] = \det[\boldsymbol{Q}] \leq \det[\boldsymbol{R_{xx}}]. \quad \square \tag{8}$$

IV. Comments

In this paper, we have developed a characterization of generalized polyphase representations (GPP). The GPP allows us a greater freedom in designing multirate systems. We studied a particular application of GPP, namely in enhancing the coding gain of transform coding systems. The advantage of using GPP was demonstrated for several inputs. Moreover the additional complexity of the new system is only slightly greater than the transform coding system, the difference being the higher number of delay elements used. We also proved several properties of the new system.

References

[1] M. Bellanger, G. Bonnerot, M. Coudreuse, "Digital filtering by polyphase network: application to sample rate alteration and filter banks," *IEEE Trans. on ASSP*, vol. 24, pp. 109-114, Apr. 1976.

[2] R. Crochiere and L. Rabiner, *Multirate Digital Signal Processing*. Englewood Cliffs, NJ: Prentice-Hall, 1983.

[3] P. P. Vaidyanathan, "Multirate digital filters, filter banks, polyphase networks, and applications: a tutorial" *Proc. IEEE,* vol. 78, no. 1, Jan. 1990.

[4] S. K. Mitra, "Some unconventional signal processing applications of multirate techniques," *Proc. ISCAS*, pp. 13-16, Singapore, June 1991.

[5] J. J. Huang and P. M. Schultheiss, "Block quantization of correlated gaussian random variables, *IEEE Trans. Comm. Syst.*, pp. 289-296, Sept. 1963.

[6] N. S. Jayant and P. Noll, *Digital Coding of Waveforms-Principles and Applications to Speech and Video.* Englewood Cliffs, NJ: Prentice-Hall, 1984.

[7] S. Kay and S. Marple, "Spectrum analysis: a modern perspective," *Proc. IEEE*, vol. 69, pp. 1380-1419, Nov. 1981.

[8] Y. Neuvo, C. Dong, and S. K. Mitra, "interpolated finite impulse response filters," *IEEE Trans. ASSP*, pp. 563-570, June 1984.

[9] T. Nguyen and P. P. Vaidyanathan, "Maximally decimated perfect-reconstruction filter banks with pairwise mirror-image analysis (and synthesis) frequency responses, *IEEE Trans. ASSP*, pp. 693-706, May 1988.

[10] P. P. Vaidyanathan, *Multirate Systems and Filter Banks..* Englewood Cliffs, NJ: Prentice-Hall, 1993.

[11] M. Vetterli, "A theory of multirate filter banks," *IEEE Trans. ASSP*, pp. 356-372, Mar. 1987.

[12] R. Horn and C. Johnson, *Matrix Analysis.* Cambridge: Cambridge Univ. Press, 1991.

Corrections on "Two Dimensional IIR Digital Notch Filter Design[1]"

Soo-Chang Pei and Chien-Cheng Tseng

There are four figures misplaced in the above paper.[1] The correction is as follows: Fig. 4(e) needs to be interchanged with Fig. 5(a), and Fig. 5(b) needs to be interchanged with Fig. 7(a).

Manuscript received September 24, 1992; revised August 18, 1993. This paper was recommended by Associate Editor Y. C. Lim.

The authors are with the Department of Electrical Engineering, National Taiwan University, Taipei, Taiwan, Republic of China.

IEEE Log Number 9405355.

[1] ____, *IEEE Trans Circuits Syst.—II*, vol. 41, no. 3, pp. 227–231, Mar. 1994.